#10

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No. **DE920000049US1**

In Re Application Of: **E.-M. Hamann et al**

Serial No.	Filing Date	Examiner	Group Art Unit
09/931,613	**August 16, 2001**	**E. P. Leroux**	**2171**

Invention: **Ensured Access To Static Objects Inside A Dynamic Token Memory**

TO THE COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on **02/24/2004**

The fee for filing this Appeal Brief is: **$330.00**

☐ A check in the amount of the fee is enclosed.

☒ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **09-0463**

Signature

Dated: **April 23, 2004**

William A. Kinnaman, Jr. - Attorney
Registration No. 27,650
IBM Corporation - MS P386
2455 South Road
Poughkeepsie, NY 12601

I certify that this document and fee is being deposited on **4/23/2004** with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature of Person Mailing Correspondence

Sandra L. Kilmer

Typed or Printed Name of Person Mailing Correspondence

cc:

IN THE U.S. PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant: E.-M. HAMANN et al.	:	Group Art Unit: 2171
Serial No.: 09/931,613	:	Examiner: Etienne Pierre Leroux
Filed: August 16, 2001	:	April 23, 2004
Confirmation No.: 3019	:	William A. Kinnaman, Jr.
Title: ENSURED ACCESS TO STATIC	:	International Business Machines Corporation
OBJECTS INSIDE A DYNAMIC TOKEN	:	2455 South Road, Mail Station P386
MEMORY	:	Poughkeepsie, NY 12601

APPLICANTS' APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants hereby submit their appeal brief in the above-identified application.

CERTIFICATE OF MAILING UNDER 37 CFR 1.8(a)

I hereby certify that this correspondence is being deposited in triplicate with the United States Postal Service as first-class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on April 23, 2004.

Sandra L. Kilmer

Sandra L. Kilmer

April 23, 2004

Date:

REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, the assignee of record.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1-10, constituting all claims pending in the application, stand rejected and are on appeal.[1] No claims have been cancelled or withdrawn.

STATUS OF AMENDMENTS

There are no unentered amendments after final rejection.

SUMMARY OF INVENTION

The invention relates to a method for managing a dynamic file system 12 (Fig. 1), especially for small handheld data carriers such as for smart cards, i.e., chipcards having their own processor. In accordance with the invention, one or more static data objects 22 (Fig. 1) are embedded in the dynamic file system 12. The static objects 22 are excluded from actions (such as management actions) performed dynamically on the dynamic file system 12. The static, embedded objects 22 may have a fixed memory address inside the dynamic file system 12 and cannot be moved to a different location by the dynamic file management functions. The static data objects 22 can be

[1] Regarding the date of the appeal, a Notice of Appeal was mailed February 17, 2004 (February 16 being a legal holiday), with a certificate of mailing under 37 CFR 1.8(a). A stamped postcard from the USPTO appears to give a receipt date for the Notice of Appeal of February 20, 2004. However, an Advisory Action mailed March 1, 2004 (paper no. 9) gives a filing (i.e., receipt) date for the Notice of Appeal of February 24, 2004, as does the online PAIR system. Although the official appeal date may thus need correction, applicants submit that they are entitled to rely on that date in determining the timeliness of this brief without an extension of time.

accessed by easy command sequences without any complex file management functions, for example by boot routines in order to check personal security-relevant data.

ISSUES

I. Whether claims 1, 5 and 8-10 were properly rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,357,005 to Devaux et al. ("Devaux").

II. Whether claims 2-4 and 6-7were properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Devaux, either alone or in combination with other cited references.

GROUPING OF CLAIMS

The claims on appeal stand or fall together; there is no grouping of claims.

ARGUMENT

Claims 1, 5 and 8-10

This group of claims on appeal contains but a single independent claim: claim 1, which reads as follows:

> 1. (Original) A method for managing a dynamic file system, comprising the step of:
>
> embedding one or more static data objects in the dynamic file system which are excluded from actions performed dynamically on the file system.

In applicants' claimed invention, therefore, one or more static data objects that are excluded from actions performed dynamically on a dynamic file system are embedded in the file system. By thus "playing around" certain static data objects, applicants are able to enjoy the advantages of a

dynamic file system (in terms of flexible memory allocation and the like), while retaining the ability to statically access certain objects (as at boot time, when the file system has not yet been loaded).[2]

Claims 1, 5 and 8-10 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,357,005 to Devaux et al. ("Devaux") (paper no. 7, ¶ 2).

Devaux discloses a system for the secure CD-ROM storage of data in which an electronic decryption microcircuit 13 is embedded in the plastic of the central region 12 of a CD-ROM 1 (Fig. 1). A chip card 3 that is inserted into a connector 23 of a CD-ROM drive 2 contains at least part of a cryptographic key, while any remaining part of the key is stored in the microcircuit 13. The CD-ROM drive 2, the chip card 3 and the microcircuit 13 interact with one another to decrypt data stored in encrypted form on the CD-ROM 1.

From this summary, it will be seen that about the only thing applicants' claimed invention and Devaux's system have in common is the concept of embedding. What they embed, however, is completely different. In applicants' claimed invention, static objects are embedded in a dynamic file system.[3] Devaux, on the other hand, embeds a physical device (microcircuit 13) in a physical substrate (CD-ROM 1), and in the central region 12 rather than the annular region 10 where the data is stored. Devaux gives no information at all about the organization of the data stored on the CD-ROM 1, much less any suggestion of a file system. For all that the patent says about the matter, CD-ROM 1 may simply store an undifferentiated set of bytes that are read sequentially from start to finish.

---

[2] The Examiner has previously responded to the statements in this paragraph with the assertion that "the features upon which applicant relies are not recited in the rejected claim(s)" (paper no. 7, page 7). Applicants fail to understand how the Examiner reaches this conclusion, especially since they are arguing the very feature—embedding static objects in a dynamic file system—recited so prominently in the main claim. And while the last sentence of this paragraph does not precisely track the actual claim recitation, the benefits recited follow almost as of course (absent some perverse implementation) from the actual claim recitation.

[3] As defined in the online encyclopedia Webopedia (http://www.webopedia.com/), a "file system" is "[t]he system that an operating system or program uses to organize and keep track of files" (hyperlinks omitted) A "file system" is thus a programming construct and not simply anything (hardware, software or whatever) that might contain a file.

Even if data on the CD-ROM were organized into a file system, it would not be a dynamic file system, since a CD-ROM is by definition read only and the stored information never changes. Further, neither microcircuit 13 nor the data stored in the microcircuit would be "embedded" in that file system. There is no chance, for example, that in the absence of some special exclusion mechanism, a person reading data off CD-ROM 1 would read the key stored in microcircuit 13. Not only is the storage area of microcircuit 13 not contiguous with that of CD-ROM 1, but the technologies used are distinct. Microcircuit 13 thus communicates with the CD-ROM drive 2 by means of an inductive antenna 14, which is entirely separate from the laser or LED that is used to read the CD-ROM itself.

In sum, Devaux teaches embedding an electronic circuit in the substrate of an optical storage medium. It does not teach embedding a static object in a dynamic file system as claimed by applicants, nor would the disclosed system be capable of functioning like the one claimed by applicants.

Claims 2-4 and 6-7

Each of these claims, which depend on claim 1, stands rejected 35 U.S.C. § 103(a) as being unpatentable over Devaux, either alone or in view of another reference. Thus, claims 3 and 4 stand rejected as being unpatentable over Devaux (paper no. 7, ¶ 4), while claim 2 stands rejected as being unpatentable over Devaux in view of U.S. Patent 5,819,252 to Benson et al. ("Benson") (paper no. 7, ¶ 4), and claims 6-7 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Devaux in view of Cepulis et al. U.S. application 2001/0042225 ("Cepulis") (paper no. 7, ¶ 6).

Suffice it to say that neither of the secondary references teaches embedding a static object in a dynamic file system as claimed by applicants. Thus, whatever relevance they may have to the secondary features for which they were cited, they do not cure the deficiency of Devaux as a primary reference.

Thus, Benson describes a method for detecting and handling an invalid use of a data structure. A data structure 30 (Fig. 2A) associated with a first (e.g., 64-bit) computing environment includes a field 30d having a stored value identifying an inaccessible address in a second (e.g., 32-bit) computing environment. The field 30d is used to detect an invalid use of the data structure in the second computing environment by a computer program 21 attempting to access memory using the inaccessible address indicated by the value contained in the field 30d.

Assuming, for the sake of argument, that Benson uses a file system (or perhaps separate 32-bit and 64-bit file systems), there is nothing to suggest any dichotomy between static objects and a dynamic file system as claimed by applicants. If Benson prohibits certain memory accesses, it is because of incompatibilities between programs and resources, not because of any attempt to access a "static" object in a dynamic file system.

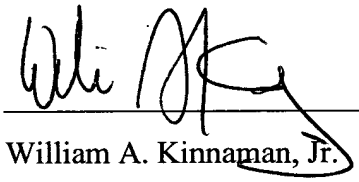
Similarly, Cepulis discloses a computer system 100 (Fig. 1) comprising a plurality of CPUs 102 including a boot-strap processor (BSP), a plurality 118 of memory modules 112, a North bridge device 106 coupling together the CPUs 102 and the memory modules 112, a master control device 140 coupled to the North bridge device via a primary expansion bus 108 and coupled to the CPUs 102 and the memory modules 112 via a serial bus (I^2C). Each CPU 102 and each memory module 112 include non-volatile memory 104 for storing an ID code uniquely identifying that CPU or memory module. A non-volatile memory device 130 coupled to the North bridge device 106 stores a failed device log 132 that includes a list of ID codes associated with a CPU 102 or memory module 112 that has failed. The master control device 140 retrieves the ID codes during initialization and provides the ID codes to the boot-strap processor BSP, which compares the retrieved ID codes with the list of ID codes in the failed device log 132 to determine if a CPU 102 or memory module 112 that is listed in the failed device log as failed is present in the computer system 100.

While the master control device 140 and boot-strap processor BSP of Cepulis may access ID codes during initialization, these "objects" are stored in a non-volatile memory device 130 and are not "embedded" in a dynamic file system as claimed by applicants. No special action is thus necessary to "exclude" such objects from actions performed on such a file system.

Conclusion

In summary, not only does Devaux fail to teach applicants' claimed invention, but the other cited references fail to teach it as well, either singly or in combination. Accordingly, applicants respectfully submit claim claims 1-10 distinguish patentably over the references cited. Applicants therefore respectfully request that the Examiner's rejection of these claims be reversed.

Respectfully submitted,
E.-M. HAMANN et al.

By 

William A. Kinnaman, Jr.
Registration No. 27,650
Phone: (845) 433-1175

WAK/wak

APPENDIX

Claims on Appeal

1. (Original) A method for managing a dynamic file system, comprising the step of:

embedding one or more static data objects in the dynamic file system which are excluded from actions performed dynamically on the file system.

2. (Original) The method of claim 1 comprising the step of:

defining an embedded static object by a memory address and a fixed size.

3. (Original) The method of claim 1 comprising the steps of:

creating an embedded static data object by specifying a predetermined storage size;

scanning memory for an available storage area large enough in size for receiving the static data object; and

allocating the storage area for the static object.

4. (Original) The method of claim 1 comprising the steps of:

creating an embedded static data object by specifying a storage area having a predetermined memory address and a predetermined storage size;

allocating the storage area if it is not being used; and

if the area is already in use, moving data using the area to a different memory location and allocating the area thereafter.

5. (Original) The method of claim 1 used for managing a file system on a chipcard.

6. (Original) The method of claim 5 comprising the step of accessing a static object in a pre-boot phase of a host system connected to the chipcard.

7. (Original) The method of claim 6 comprising the step of storing security-relevant data in a static object.

8. (Original) A chipcard comprising a dynamic file system managed by the method of claim 1.

9. (Original) A computer operating system program for execution in a data processing system comprising computer program code portions for performing respective steps of the method of claim 1 when the code portions are executed on the data processing system.

10. (Original) A computer program product stored on a computer usable medium comprising computer readable program means for causing a computer to perform the method of claim 1 when the program product is executed on the computer.